18.453 lecture 4 - 5

Lecture plan:

1. non-bipartite matchings.

2. Tutte-Berge

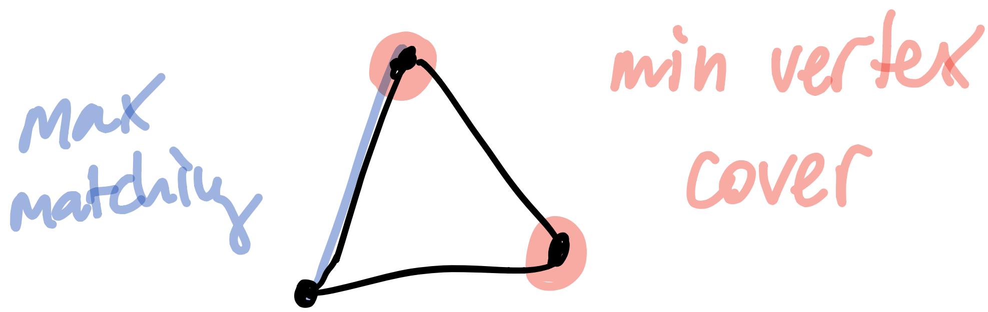
3. Algorithmic proof (Edmonds' alg)

* might not finish!

Non-bipartite Matching

- Given $G = (V, E)$;
  do <u>not</u> assume bipartite.
- Want maximum matching M in G.
- König's theorem doesn't hold:

  max matching $\lneq$ min vertex cover.



- Recall from lecture 1: instead, duality w/ obstructions based on parity.

Tutte-Berge Formula

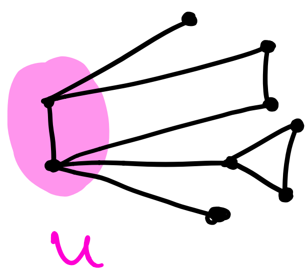
Given $U \subseteq V$,

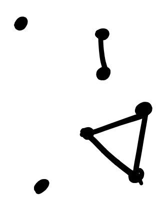
Def

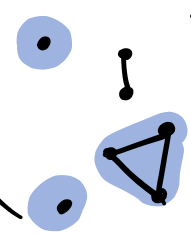
$G \setminus U :=$ G after deleting U & all adjacent edges.

$o(G \setminus U) :=$ # odd connected components in $G \setminus U$

E.g.

$G =$ 

$G \setminus U =$ 

$o(G \setminus U) = o\Big($  $\Big) = 3$

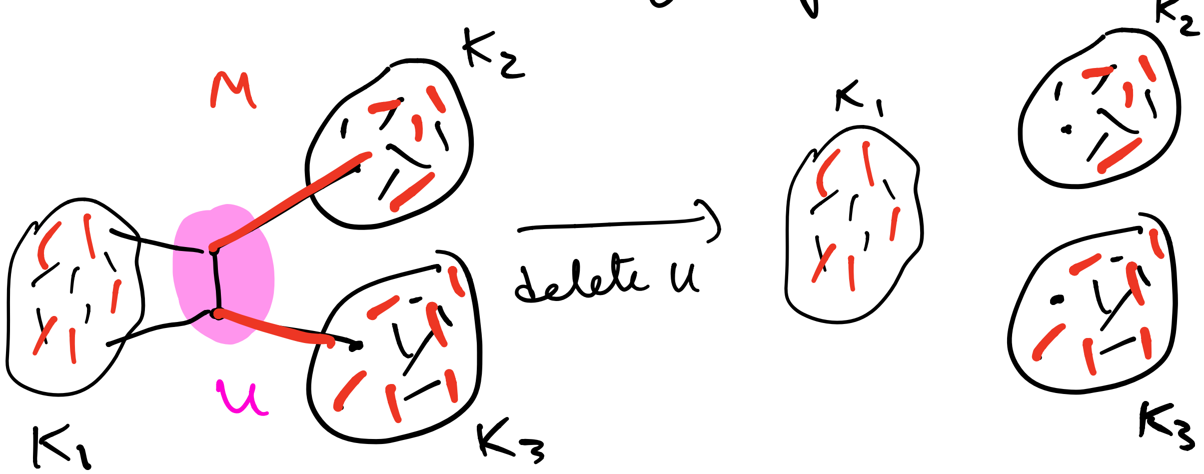
Thm (Tutte-Berge Formula):

$$\max_{\text{matching } M} \underbrace{|M|}_{\#\text{edges}} = \min_{U\subseteq V} \underbrace{\tfrac{1}{2}}_{\text{was missing in lec 1}}\Big(|V| + |U| - o(G\setminus V)\Big)$$

Pf ($\leq$) i.e. "<u>weak duality</u>"

- Deleting U deletes $\leq |U|$ edges of M.

How many left over?



- # left over is at most

$$\sum_{i=1}^{3} \left\lfloor \frac{|K_i|}{2} \right\rfloor .$$

- Thus, if $K_1 \ldots K_k$ are connected components of $G \setminus U$,

$$* \quad |M| \leq |U| + \sum_{i=1}^{k} \left\lfloor \frac{|K_i|}{2} \right\rfloor .$$

- Can rewrite:

$$\left\lfloor \frac{|K_i|}{2} \right\rfloor = \begin{cases} \frac{|K_i|}{2} & \text{if } |K_i| \text{ even} \\ \frac{|K_i|-1}{2} & \text{else.} \end{cases}$$

thus $**$

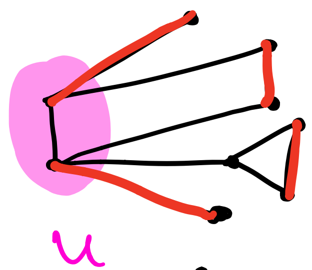
$$\sum_{i=1}^{k} \left\lfloor \frac{|K_i|}{2} \right\rfloor = \sum_{i=1}^{k} \frac{|K_i|}{2} - \frac{1}{2} o(G \setminus U).$$

$$= \frac{|V| - |U| - o(G \setminus U)}{2}.$$

• Plugging ** into * gives

$$|M| \le \frac{1}{2}\left(|V| + |U| - o(G\setminus U)\right) \quad \square$$

E.g.

G = 

$|M| = 4$, $\quad \frac{1}{2}\left(|V| + |U| - o(G\setminus U)\right) = \frac{1}{2}(9 + 2 - 3) = 4.$

Proof of $\geq$??

• Beautiful algorithm due to Edmonds.

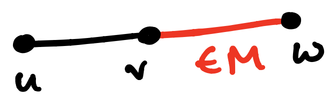
• challenge: though still true that

look carefully at proof from lec 2!

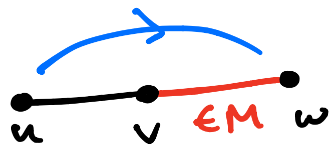
$\boxed{M \text{ maximum} \iff \text{no aug. path w.r.t } M,}$

finding the paths is hard.

- Why? Natural approach repeats vertices.

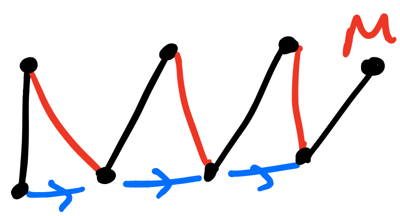
<u>Natural approach</u>: whenever you see 

, add directed edge uw:



.

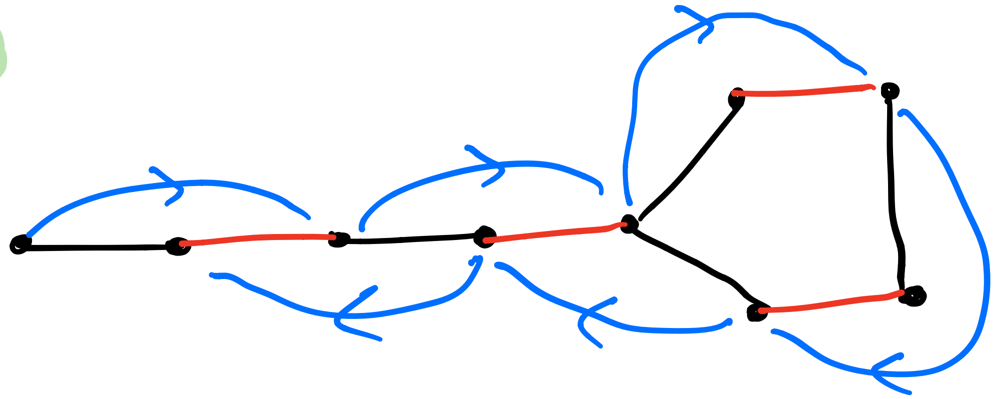
<u>E.g.</u>



Then, <u>start</u> at exposed vertex & look for vertex adjacent to an exposed vertex in blue digraph.

<u>Problem</u>: can lead to repeated vertices.

E.g.



when we first repeat, have found a

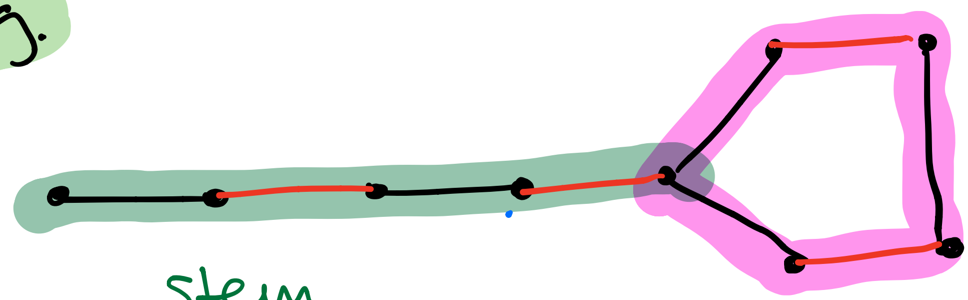
flower (with respect to M):

Stem: even-length alternating path from exposed u to vertex v

+

Blossom: odd length cycle intersecting stem in v, alternating except for edges incident to v.

E.g.



Algorithm idea:

At each step, have matching M.

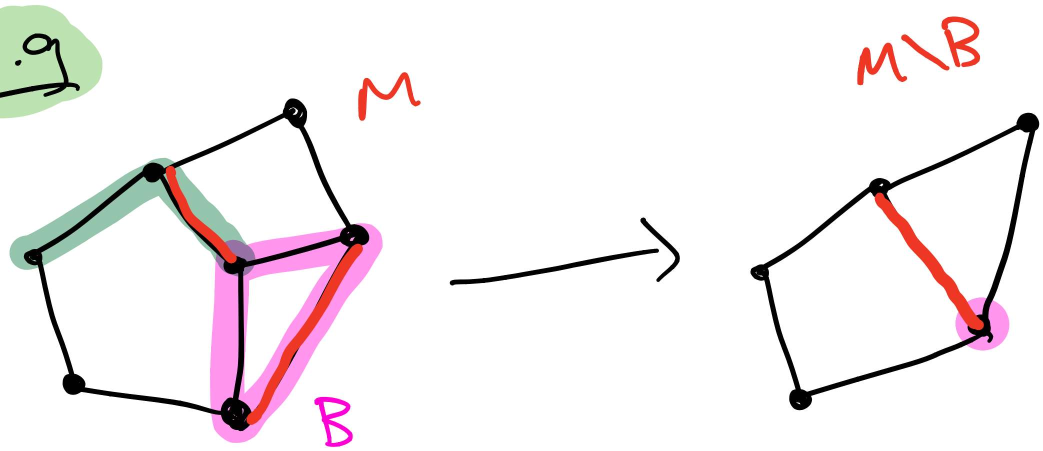
- find aug. path or flower w.r.t M or show neither exists.
- If neither exists, matching is maximum.
- if aug. path, augment & repeat.

- if <u>flower</u>, let B be blossom.

Create graph G/B (not $G \setminus B$) called <u>contraction</u> where

1. B shrunk to single vertex b.
2. edges (u, v) for $u \notin B$, $v \in B$ replaced by $(u, b) \in G/B$.

<u>E.g</u>



G $\qquad$ G/B.

<u>Note</u>: is matching M/B in G/B

and $|M| - |M/B| = \frac{|B|-1}{2}$.

(i.e. # edges of M in B).

<u>**Crucial Theorem**</u>: Let B be a blossom w.r.t. M. Then

$\boxed{M \text{ max. matching in } G}$

$\Longleftrightarrow$

$\boxed{M \setminus B \text{ max. matching in } G \setminus B.}$

Proof will be <u>algorithmic</u>:

If bigger matching in $G \setminus B$ than

$M \setminus B$, can use it to find bigger matching in G than M.

Theorem → Algorithm: recursion!

Assuming we can find either aug path or blossom, can recurse to increase size of $M \setminus B$ in $G \setminus B$.

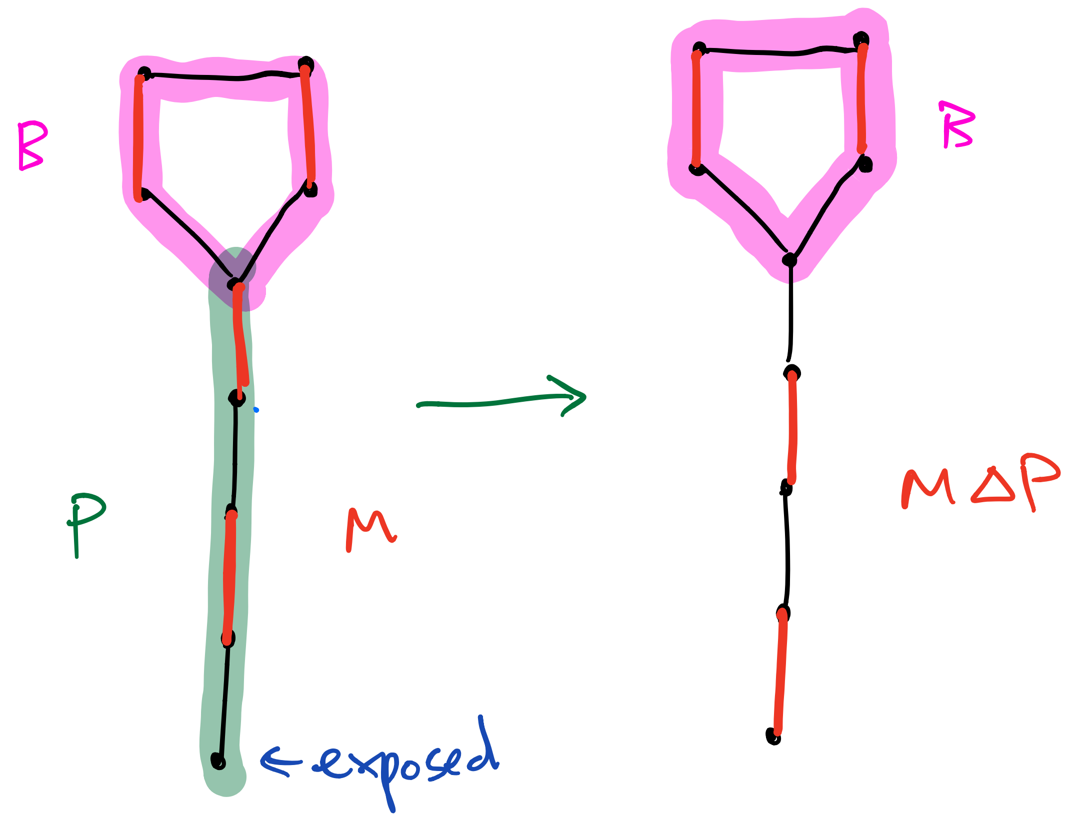
- if not possible, M maximum.
- else, use new matching in $G \setminus B$ to increase M.

Proof of Crucial Theorem:

0. W.L.O.G. assumption:

B has empty stem P!

Why W.L.O.G? If P nonempty, look at $M \Delta P$.



- $M \Delta P$ has empty stem & blossom P.

- Proving theorem for $M \Delta P$ also proves for M:

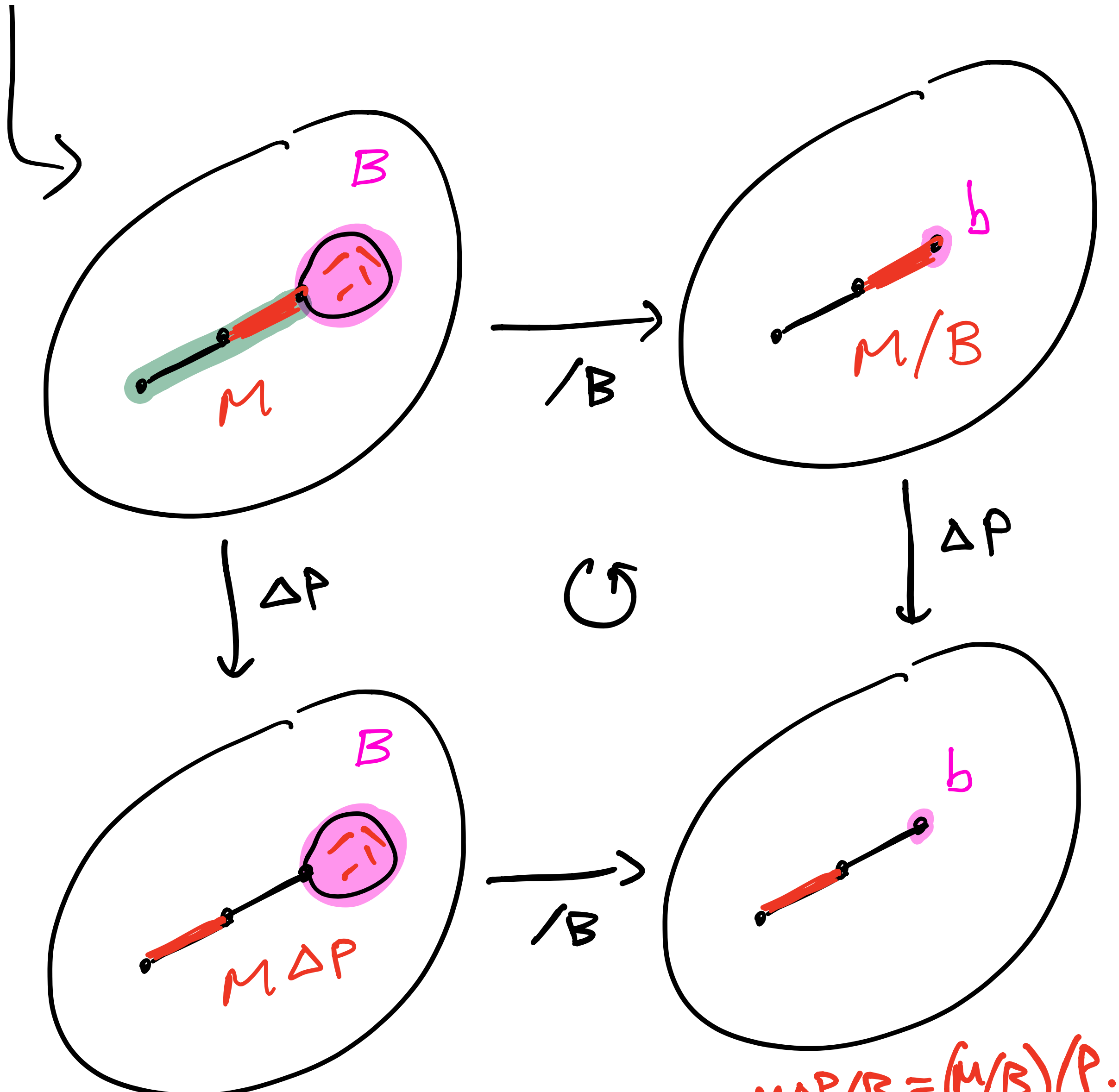
M maximum in G

P alternating $\Longleftrightarrow$ $M \Delta P$ maximum in G

theorem $\Longleftrightarrow$ $M \Delta P / B$ max in G/B

$M \Delta P / B = (M/B) \Delta P$ $\Longleftrightarrow$ $(M/B) \Delta P$ max in G/B

P alternating $\Longleftrightarrow$ (M/B) max in G/B.



$M\Delta P/B = (M/B)/P.$

Finally, start proof. Recall <u>Thm</u>:

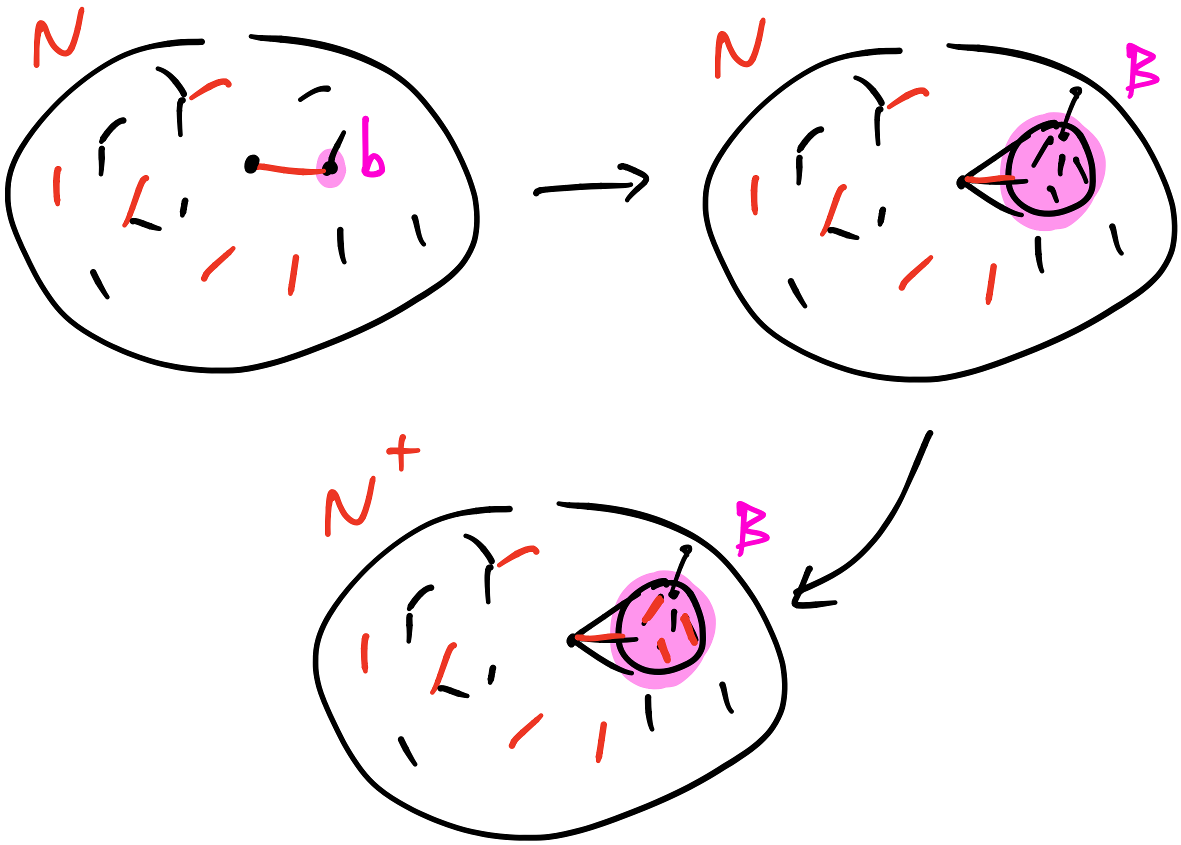
$$\boxed{M \text{ max. in } G} \iff \boxed{M\backslash B \text{ max in } G\backslash B}$$

1. ($\Rightarrow$): Suppose N is matching G/B larger than M/B.

- pull back N to matching in G; is incident to ≤ 1 vertex of B.

- Expand to matching $N^\uparrow$ in G by adjoining

$\frac{1}{2}(|B|-1)$ edges to match remaining vertices of B.



$|N^+|$ exceeds $|M|$ by same amt. $|N|$ exceeds $|M/B|$.

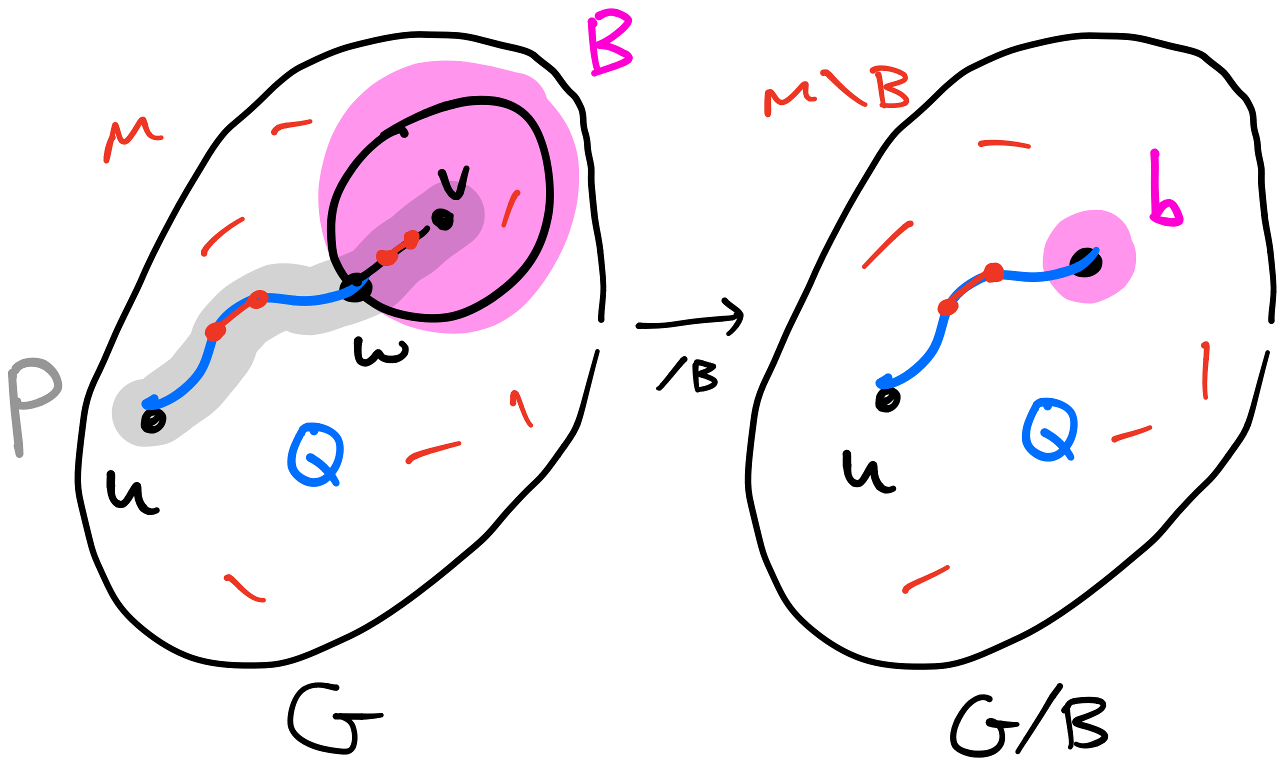
2. ($\Leftarrow$)

Contrapositive: if M not max, M/B not max.

Suppose M not max in G.

- then $\exists$ aug path P between exposed verts u, v.
- wlog $u \notin B$,
  b/c B has 1 exposed vertex. (empty stem)
- $w := \begin{cases} \text{first vertex of } P \text{ in } B \text{ (starting at } u) \\ v \text{ if } P, B \text{ share no vertices.} \end{cases}$
- $Q :=$ part of P between u, w.
- Q augmentg path in M/B

b/c b exposed in M/B.



if P, B vertex disjoint, v still exposed.

Augmenting M/B along Q ⇒

Thus M/B not maximum. □

Note! thm doesn't say

maximum matching M^* in G/B

$\rightsquigarrow$ max matching $\hat{M}$ in G

by adding $\frac{|B|-1}{2}$ edges

from B to M^*.

<u>Ex.</u> find example of this;

explain why no contradiction.

<u>Finally, ready to give algorithm.</u>

Lecture 5

Plan:

1. finish Edmonds' nonbipartite matching alg.
2. Use to prove Tutte's theorem.

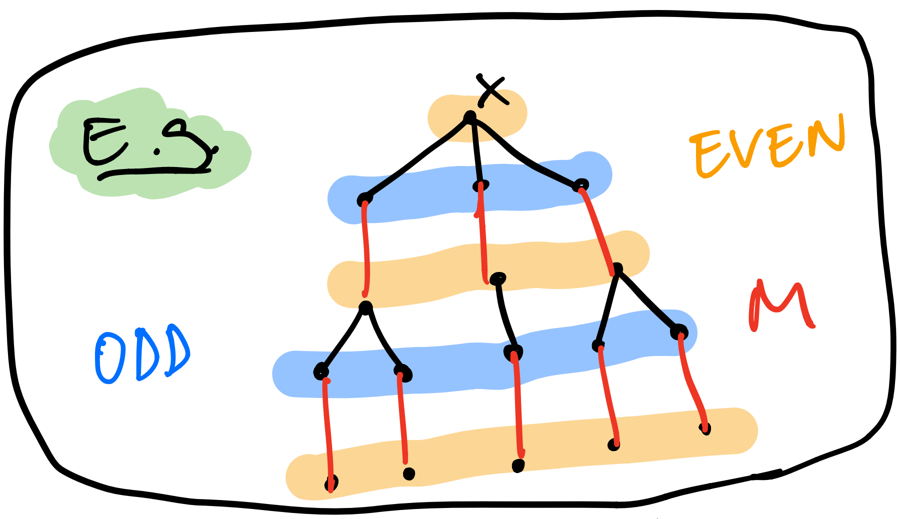
Edmonds' Algorithm

Given M, find aug path / flower.

- Label exposed vertices EVEN;

Keep others unlabelled initially.
(eventually will label others ODD/EVEN).

- Maintain alternating forest: graph in which each connected component is alternating tree (AT) i.e. tree w/ paths to root
  - i) alternating w.r.t M.
  - ii) alternating b/w ODD & EVEN.



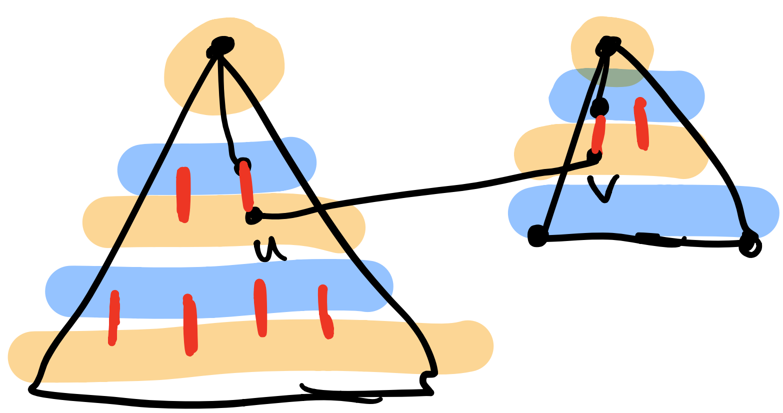
- Process EVEN vertices one at a time. If

currently processing u,
cases based on neighbors of u.

(a) If edge (u,v) with v unlabelled, label v ODD. v not exposed (b/c else v EVEN); So label v's mate w EVEN. Add (u,v), (v,w) to u's AT.

(b) if $\exists$ edge (u,v) s.t. v EVEN and v belongs to different AT than u:

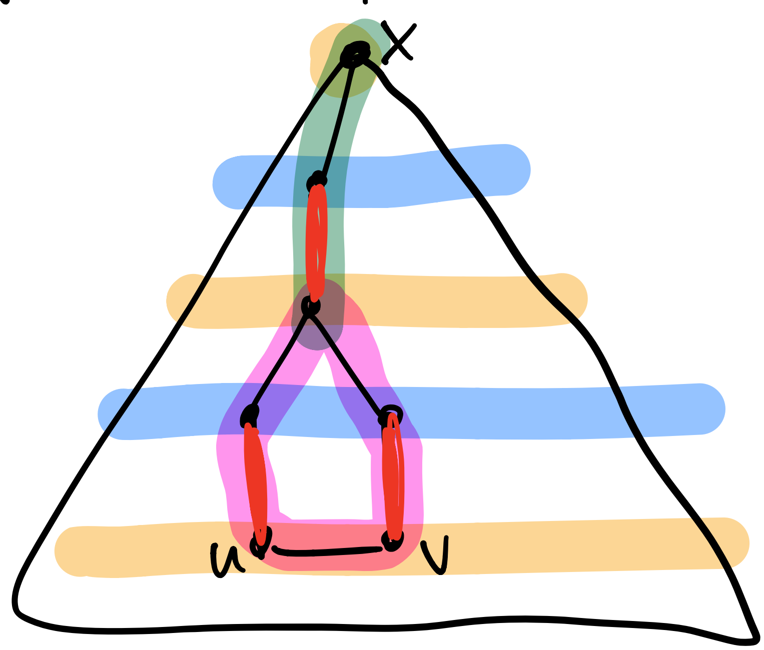
Then $\exists$ aug path between the roots!



have found aug path; increase M, start over with new M.

ⓒ If is edge labeled (u,v)
with v labeled <u>EVEN</u>
& v in same AT as u,

then: two paths from
u, v to (exposed) root x
form a flower.



Shrink to $G \backslash B$, recursively find max. matching in $G \backslash B$, use it to increase M using the crucial theorem. Start over w/ new M.

Correctness: Suppose none of a, b, c apply anymore for the EVEN vertices.

Claim: Current matching M_K

is max in current $G_k = (V_k, E_k)$

$G_k := \underbrace{G/B_1/B_2/\ldots}_{G_i}/B_k$

for B_i blossom in G_{i-1}.

Proof of Claim: Consider $U = \text{ODD}$ and consider the upper bound from Tutte-Berge for G',

$$|M'| \le \frac{1}{2}\left[|V'| + |U| - o(G' \setminus U)\right].$$

- No edges b/w EVEN vertices (else (b) or (c) applies).
  & no edges b/w EVEN & unlabelled (else (a) applies).
- Thus, EVEN are singleton components in $G' \setminus U$,

so $o(G' \setminus \underline{ODD}) = |\underline{EVEN}|$ *

- Further, all <u>unlabelled</u> vertices matched, so

$$** \begin{cases} |M'| = |ODD| + \frac{1}{2}(|V'| - |ODD| - |EVEN|) \\ \quad = \frac{1}{2}(|V'| + |ODD| - |EVEN|) \end{cases}$$

- Plug * into ** :

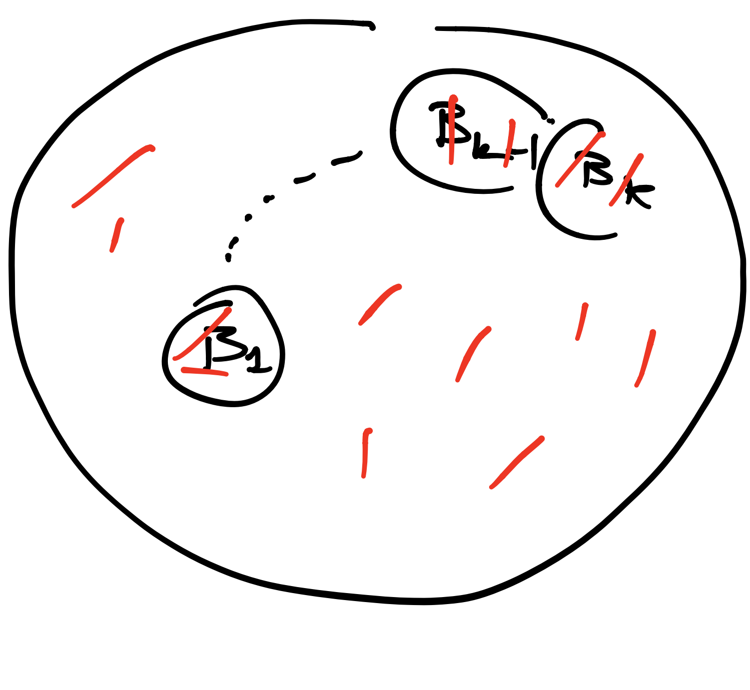
$$|M'| = \frac{1}{2}(|V'| + |ODD| - |EVEN|)$$

$$= \frac{1}{2}(|V'| + |ODD| - o(G' \setminus ODD)).$$

<u>Tutte Berge (upper bound)</u> $\Rightarrow$

M' max in G'. Applying crucial theorem repeatedly for $B_k B_{k-1} \ldots B_1$ shows algorithm constructs maximum matching in G.

because B_i was blossom in G_{i-1}



Running time.

- Algorithm performs $\leq \frac{n}{2}$ augmentations of matching "outer loop"
- between two augmentations, "inner loop" shrinks blossom $\leq \frac{n}{2}$ times (shrinks by ≥ 2 vertices).
- Time to construct AT is $O(m)$, $m := |E|$.

$\boxed{\text{So overall, } O(n^2 m).}$

Proof of Tutte-Berge $\geq$

We've argued TB holds for graph G_k for which alg. terminates.

- Recall G_i obtained from shrinking blossoms $B_1, \ldots, B_i$, let M_i corresp. matching. $G_0 := G$.

- We saw TB holds for G_k, i.e.

$$|M_k| = \frac{1}{2}\left(|V_k| + |U| - o(G_k \setminus U)\right)$$

where $U = \text{ODD}$,

b/c $G_k \setminus \text{ODD} = \text{EVEN}$;

singletons components.

- Unshrink B_i, one at a time, induct backwards.

$$\boxed{\text{In step } G_i \longrightarrow G_{i-1}:}$$

(i) $|V_{i-1}| = |V_i| + |B_i| - 1$

bitself.

and

$$|M_{i-1}| = |M_i| + \frac{1}{2}(|B_i| - 1).$$

(ii) Unshrinking B_i adds even $(|B_i|-1)$ vertices to some C.C. of $G_i \setminus U$, so # odd/even components stays same.

i.e. $\boxed{o(G_i \setminus U) = o(G_{i-1} \setminus U)}$.

(iii) Using this, when $i \leftarrow i-1$

the RHS & LHS of

$$|M_i| = \frac{1}{2}\left(|V_i| + |U| - o(G_i \setminus U)\right)$$

increase by $\frac{1}{2}(|B_i| - 1)$.

By induction,

$$\boxed{|M_0| = \frac{1}{2}\left(|V_0| + |U| - o(G_0 \setminus U)\right).}$$

$\square$

Corollary of Tutte-Berge:

G has p.m. iff

$\forall U, \ o(G \setminus U) \leq |U|$.

This is called

Tutte's matching thm.